VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE

Brno, 2016

Péter Ledniczky

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ
DEPARTMENT OF TELECOMMUNICATIONS

WEBOVÁ APLIKACE PRO PROHLEDÁVÁNÍ ZMÍNEK O PRODUKTECH Z INTERNETOVÝCH PORTÁLŮ
WEB APPLICATION FOR SEARCHING FOR DOCUMENTS RELATED TO GIVEN PRODUCT

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **Péter Ledniczky**
AUTHOR

VEDOUCÍ PRÁCE — **doc. Ing. Radim Burget, Ph.D.**
SUPERVISOR

BRNO 2016

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Bakalářská práce

bakalářský studijní obor **Teleinformatika**
Ústav telekomunikací

Student: Péter Ledniczky ***ID:*** 146888
Ročník: 3 ***Akademický rok:*** 2015/16

NÁZEV TÉMATU:

Webová aplikace pro prohledávání zmínek o produktech z internetových portálů

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s problematikou návrhu webových aplikací. Navrhněte aplikaci, která bude vyhledávat z jiných stránek záznamy, kde se vyskytuje zmínka o hledaném produktu. Navrhněte vhodné grafy a HTML stránky, které budou zobrazovat trend vývoje nálady ohledně daného produktu.

DOPORUČENÁ LITERATURA:

[1] DUCKETT, J. HTML and CSS: Design and Build Websites. John Wiley & Sons, 2011.

[2] NIXON, R. Learning PHP, MySQL, JavaScript, and CSS: A step-by-step guide to creating dynamic websites. " O'Reilly Media, Inc.", 2012.

Termín zadání: 1.2.2016 ***Termín odevzdání:*** 1.6.2016

Vedoucí práce: doc. Ing. Radim Burget, Ph.D.
Konzultant bakalářské práce:

doc. Ing. Jiří Mišurec, CSc., *předseda oborové rady*

ABSTRAKT

Cieľom tohto projektu je vytvorenie programu, ktorý bude automaticky zbierať dostupný textový obsah z internetu a následne v ňom vyhľadá vopred zadané kľúčové slová. Na základe ich výskytu vykoná analýzu náladového indexu textu. Výsledky hodnotenia sú následne prezentované prostredníctvom grafov. Práca je vyhotovená s použitím technológií HTML, CSS, JavaScript, PHP a SQL.

KĽÚČOVÉ SLOVÁ

Crawler, web, kľúčové slovo, HTML, CSS, JavaScript, PHP, SQL, Smarty, Bootstrap, RSS

ABSTRACT

The aim of this project is to create a web aplication that will automatically collect the available text content from the Internet. Afterwards it looks for the predefined keywords and according to their occurrence it analyzes whimsical text index. The evaluation results are then presented through graphs. Work is done using HTML, CSS, JavaScript, PHP and SQL.

KEYWORDS

Crawler, web, keyword, HTML, CSS, JavaScript, PHP, SQL, Smarty, Bootstrap, RSS

LEDNICZKY, P. *Webová aplikace pro prohledávání zmínek o produktech z internetových portálů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací, 2015. 41 s. Bakalářská práce. Vedoucí práce: doc. Ing. Radim Burget, Ph.D.

PREHLÁSENIE

Prehlasujem, že som svoju bakalársku prácu na tému „Webová aplikace pro prohledávání zmínek o produktech z internetových portálů“ vypracoval samostatne pod vedením vedúceho bakalárskej práce, využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej bakalárskej práce ďalej prehlasujem, že v súvislosti s vytvorením tejto bakalárskej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/nebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona č. 121/2000 Sb., o právu autorském, o právoch súvisejúcich s právom autorským a o zmeně niektorých zákonov (autorský zákon), vo znení neskorších predpisov, vrátane možných trestnoprávnych dôsledkov vyplývajúcich z ustanovenia časti druhé, hlavy VI. diel 4 Trestného zákoníka č. 40/2009 Sb.

V Brne dňa ....

(podpis autora)

POĎAKOVANIE

Rád by som poďakoval vedúcemu bakalárskej práce pánovi doc. Ing. Radimu Burgetovi, Ph.D. za odborné vedenie, konzultácie, trpezlivosť a podnetné návrhy k práci.

V Brne dňa ....

(podpis autora)

Faculty of Electrical Engineering
and Communication
Brno University of Technology
Purkynova 118, CZ-61200 Brno
Czech Republic
http://www.six.feec.vutbr.cz

POĎAKOVANIE

Výzkum popsaný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno ....
(podpis autora)

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

OBSAH

ZOZNAM OBRÁZKOV

ÚVOD

Bakalárska práca popisuje program, ktorý vyhľadáva požadované informácie a ďalej ich spracováva. Tiež umožňuje užívateľom zvolenie vlastných zdrojov, z ktorých by chceli informácie čerpať, a z ktorých program bude následne články odoberať. Užívateľ si takisto bude môcť navoliť vlastné kľúčové slová, ktoré zodpovedajú problematike, o ktorú sa zaujíma. Tieto slová bude následne program vyhľadávať v uložených článkoch a ďalej analyzovať a vyhodnocovať.

V prvom rade práca popisuje prínosu programu, potom sú charakterizované využité webové technológie, ďalej sa venuje jednotlivým častiam vývoja a funkčnosti programu. Je popísaný program z užívateľského hľadiska, následne sú ukázané výstupy programu a návrhy pre využitie a v poslednej rade práca obsahuje možné návrhy na získavania a hodnotenia článkov. V závere sú zhrnuté dosiahnuté výsledky.

Cieľom bakalárskej práce je vytvoriť program, ktorý uľahčí užívateľom prístup k informáciám, ktoré ho zaujímajú a následne tieto informácie skúmať. Ďalším zámerom je navrhnúť primerané formy aplikácie programu.

Táto práca rieši problém veľkého počtu informácií na internete. Človek častokrát nemá čas všetky tieto informácie filtrovať, čítať a analyzovať. Presne tieto činnosti zaňho vyrieši môj program. Vyfiltruje len informácie, ktoré užívateľa skutočne zaujímajú a vyčíta z nich dôležité údaje, na ktorých základe článok zanalyzuje a priradí mu príslušnú hodnotu, ktorá predstavuje jeho náladu. Nazbierané informácie užívateľovi prehľadne zobrazí prostredníctvom grafov.

Hlavným prínosom práce je spracovávanie zozbieraných dát za dlhšie časové obdobie. A hlavne vhodná aplikácia tohto programu v rôznych oblastiach. V práci sa zameriavam najmä na aplikácie v politike a ekonomike, avšak existuje široké spektrum oblastí, v ktorých je možné tento program využiť, záleží len na užívateľovi, o aký typ aplikácie analýz má záujem.

1 INTERNET AKO ZDROJ INFORMÁCIÍ

1.1 Prínos programu pre užívateľov

Internet je fenoménom dnešnej doby, pre mnohých je najmä komunikačným kanálom, ktorý pomyselne skracuje vzdialenosť medzi ľuďmi, ale rovnako je vnímaný aj ako náhrada rôznych zdrojov informácií, ako sú noviny, časopisy, knihy, televízia a podobne. Navyše tieto informácie sprostredkováva novým komfortným spôsobom z pohodlia ich domova a navyše okamžite a kedykoľvek, keď ich človek práve potrebuje. Na druhej strane, sa niekedy stretávame s tým, že internet sprístupňuje ľuďom až priveľké množstvo informácií rôzneho druhu. Pri takomto množstve informácií je občas veľmi náročné zorientovať sa a vybrať z nich tie podstatné, o ktoré má človek skutočne záujem.

Cieľom mojej práce je vyriešiť práve takéto problémy, ktoré prichádzajú s dnešnou dobou. Teda poskytnúť ľuďom službu, ktorá im pomôže lepšie sa v informáciách zorientovať, triediť ich a vyfiltrovať z nich len tie, ktoré sú pre nich naozaj dôležité. Navyše užívateľom ponúka aj ďalšie možnosti, ako je analýza a štatistické spracovávanie zozbieraných dát. V podstate môj program umožňuje, že užívateľ nemusí články ani prečítať na to, aby získal zbežný prehľad o danej problematike a o jej vývoji v určitom časovom horizonte. Zámerom práce je teda zjednodušenie práce s veľkým množstvom informácií prostredníctvom internetu.

Zmyslom práce je taktiež program aplikovať na rôzne oblasti záujmu jednotlivcov, ako aj skupín so spoločným záujmom. Aplikácia sa týka hlavne výstupov programu vo forme štatisticky spracovaných dát a tiež grafov vytvorených na ich základe. Tieto aplikácie sa týkajú takmer všetkých oblastí každodenného života okolo nás.

1.2 Webová aplikácia pre prehľadávanie stránok

Program, ktorým sa táto práca zaoberá sa nazýva Web crawler. Je to program alebo automatický skript, ktorý systematicky prehľadáva web, sťahuje súbory a indexuje ich. Indexovanie súborov znamená ich spracovávanie a ukladanie do databázy. Proces tohto prehľadávania sa nazýva Web crawling. Rozhodol som sa pre formu webovú stránku, druhou možnosťou bola forma aplikácie. Webovú stránku som zvolil z dôvodu, že si myslím, že prístup k internetu je v dnešnej dobe dostupný už takmer každému, je veľmi jednoducho ovládateľný a prácu s ním zvládne skutočne ktokoľvek. Ďalej u mňa zavážila rýchlosť internetu, ktorá je pre užívateľa pri získavaní informácií a štatistík veľmi dôležitá. Na rozdiel od aplikácie, nie je potrebné si nič inštalovať ani aktualizovať a pre všetky operačné systému je webová stránka rovnaká, takže odpadá aj problém s rôznymi verziami a otázkou ich vzájomnej kompatibility.

Ďalšou významnou časťou sú kľúčové slová. Celou podstatou web crawleru sú práve oni, ktoré užívateľ zadá do programu, a ktorými určí problematiku, ktorou sa chce zaoberať. Web crawler vlastne zaisťuje to, že tieto kľúčové slová vyhľadáva v článkoch, ktoré boli stiahnuté a uložené do databázy. Program ich až následne analyzuje a vyhodnocuje.

2 TECHNOLÓGIE

Webové stránky a aplikácie sa v dnešnej dobe najčastejšie realizujú pomocou skriptovacieho jazyka PHP [5] alebo frameworku ASP.NET písaného najčastejšie v jazyku C#. Pre tento projekt bol vybraný jazyk PHP z dôvodu jednoduchosti a absencie nutnosti kompilácie kódu.

2.1 Skriptovací jazyk PHP

Skriptovací jazyk PHP je najrozšírenejším jazykom na tvorbu webových stránok. Nie je kompilovaný, ale len interpretovaný, čo prináša rýchlejší vývoj a ladenie aplikácie. O interpretovanie sa stará software Apache HTTP Server, ktorý je multiplatformový, a tak je možné PHP skripty spúšťať prakticky na všetkých bežných platformách. PHP ponúka možnosť programovať ako s použitím tried a objektov, rovnako aj písanie lineárneho nezapuzdreného kódu. Vďaka veľkej užívateľskej základni je k dispozícii množstvo PHP frameworkov, ktoré môžu uľahčiť prácu s vývojom. Dokumentácia k tomu jazyku je dostupná na www.php.net [5].

2.2 Framework Smarty

Jeden z veľmi užitočných PHP frameworkov je framework Smarty. Je primárne určený na zjednodušenie práce s vykresľovaním dát. Odstraňuje nutnosť kontroly toho, či je premenná, ktorú sa programátor snaží vypísať, definovaná. Ďalej tiež obsahuje množstvo funkcií pre prácu s dátami. Súbory v jazyku Smarty sú následne kompilované do PHP a tak nevzniká negatívny dopad na rýchlosť načítania stránky. Dokumentácia k tomu jazyku je dostupná na www.smarty.net [6].

2.3 Databáza MySQL

MySQL je podobne ako PHP bezpochyby najpoužívanejšou databázou slúžiacou pre tvorbu webových stránok. Jedná sa o relačnú databázu, ktorej komunikácia je realizovaná jazykom SQL. Takisto je multiplatformová a voľne šíriteľná. Dokumentácia k tomu jazyku je dostupná na www.mysql.com [7].

2.4 HTML a CSS

HTML alebo Hypertext Markup Language je značkovací jazyk pre vizualizáciu výstupných dát. HTML je široko podporovaný, neustále vyvíjaný a prakticky jediný jazyk, ktorý slúži k týmto účelom. Jeho vnútorná štruktúra pripomína štruktúru XML, obsah je uzavretý medzi značkami (tagmi), ktoré potom určujú výsledný vzhľad jednotlivých blokov. CSS (Cascading Style Sheets) je potom jazyk popisujúci spôsob zobrazenia jednotlivých elementov kódu HTML. Ten môže byť písaný priamo do HTML kódu alebo do samostatného súboru. Dokumentácia k tomu jazyku je dostupná na www.w3schools.com/html [3] a www.w3schools.com/css [4].

2.5 JavaScript

JavaScript je ďalším z interpretovaných jazykov používaný pre tvorbu webových stránok. Slúži prevažne na ovládanie prvkov HTML kódu, umožňuje tvorbu interaktívnych častí stránky. Na rozdiel od PHP je JavaScript interpretovaný v prehliadači klienta, čo prináša množstvo výhod aj nevýhod. Výhody spočívajú v možnosti jednoduchého ladenia skriptu a v práci s udalosťami, ako napríklad kliknutie myšou. Nevýhodou je potom možné spomalenie webovej stránky, či priamo jej zastavenie (v prípade nesprávne navrhnutého neoptimalizovaného kódu). Vzhľadom k tomu, že klient môže nahliadnuť do zdrojového kódu sa výrazne neodporúča používanie JavaScriptu napríklad pre prácu s databázou, kde by klient teoreticky mohol získať prístup k celej databáze.

JavaScript ponúka niekoľko rôznych frameworkov, pričom pre tento projekt bol vybraný Framework jQuery [13], ktorý uľahčuje prácu s HTML elementmi a obsahuje množstvo funkcií, ktoré samotný JavaScript neobsahuje. Dokumentácia k tomu jazyku je dostupná na www.w3schools.com/js [9].

3 REALIZÁCIA A FUNKČNOSTI PROGRAMU

Táto kapitola sa zaoberá web crawlerom po technickej strane. Detailne sa venuje jednotlivým častiam vývoja a funkčnosti programu. Začína od štruktúry samotného systému a využití jednotlivých webových technológií na jeho tvorbu a ďalej popíšem konkrétne prvky, ako je práca so zdrojmi, s kľúčovými slovami, s databázou ako sú využité CRON úlohy, ako je vykreslený obsah a nakoniec je znázornená funkčnosť systému.

3.1 Realizácia programu

Program je realizovaný prostredníctvom webovej stránky. Túto možnosť som zvolil z viacerých dôvodov, jedným z nich je aj dnešná čoraz širšia dostupnosť internetu. Ako som už spomínal, druhou možnosťou by bolo sprístupniť program formou aplikácie. Hlavnú nevýhodu vidím v množstve rôznych dostupných operačných systémov, ako pri počítačoch (Windows, Mac OS), tak aj pri mobilných telefónoch (iOS, Android, Symbian, Windows Phone, Firefox OS). Jednotlivé operačné systémy majú aj rôzne verzie. Napríklad pri Windowse sú používané verzie XP, 7, 8 a 10, a navyše rozlišujeme 32 a 64bit verzie. Dokonca jednotlivé verzie nemusia byť medzi sebou ani kompatibilné. Pre každý systém by teda musela byť vytvorená samostatná aplikácia, aby mohla byť využívaná skutočne každým.

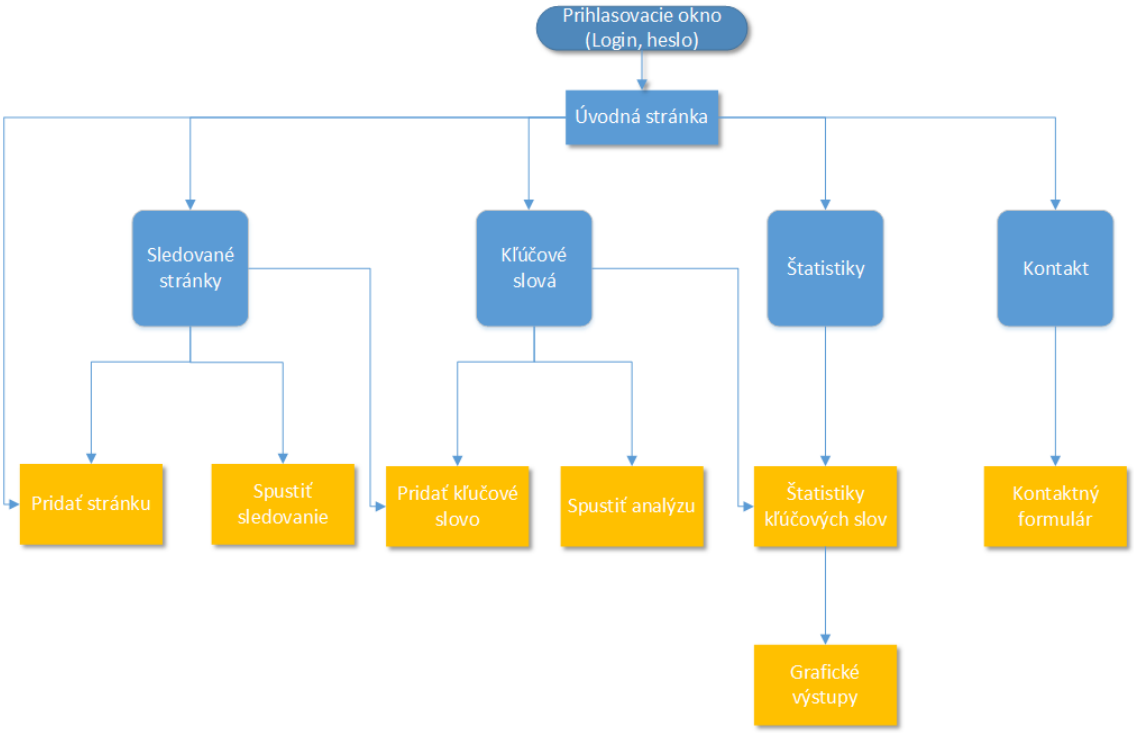
Ďalej by som chcel vyzdvihnúť ľahkú dostupnosť internetu takmer každému, preto je cieľová skupina programu pomerne rozsiahla. Množstvo ľudí si v dnešnej dobe hľadá rôzne informácie prostredníctvom internetu. Tam sa však často stretávajú aj s hromadou údajov, ktoré nepatria do ich oblasti záujmu. Tieto informácie musia vytriediť od tých podstatných, a práve túto prácu za nich dokáže vykonať web crawler, ktorý pomocou kľúčových slov vyfiltruje len tie správy, ktorým užívateľ chce skutočne venovať pozornosť a nezahlcuje ho nežiadanými údajmi.

Jednou z ďalších výhod programu vo forme webovej stránky je, že si ju užívateľ nemusí stiahnuť do počítača respektíve mobilného telefónu, či tabletu ako je to u aplikácií. Webovú stránku si jednoducho otvorí vo svojom prehliadači a okamžite môže začať s programom pracovať. Pri aplikácii je ďalej potrebné dbať o pravidelné aktualizácie, aby program pracoval správne, na webovej stránke sú tieto činnosti zaisťované automaticky.

3.2 Kostra programu

Jadro tohto programu je naprogramované v jazyku PHP [5] s využitím frameworku SMARTY [6] . Ďalej sa využíva MySQL databázu [7], do ktorej PHP ukladá záznamy o jednotlivých kľúčových slovách, zaznamenaných článkoch a hodnotení týchto článkov na základe pridelených kľúčových slov. Užívateľské prostredie samotné je naprogramované v jazyku HTML5 [3], CSS [4] s užitím jeho doplnkov Bootstrap [10] a Font-Awesome [12], ďalej tiež využívam JavaScript a jeho framework jQuery [13].

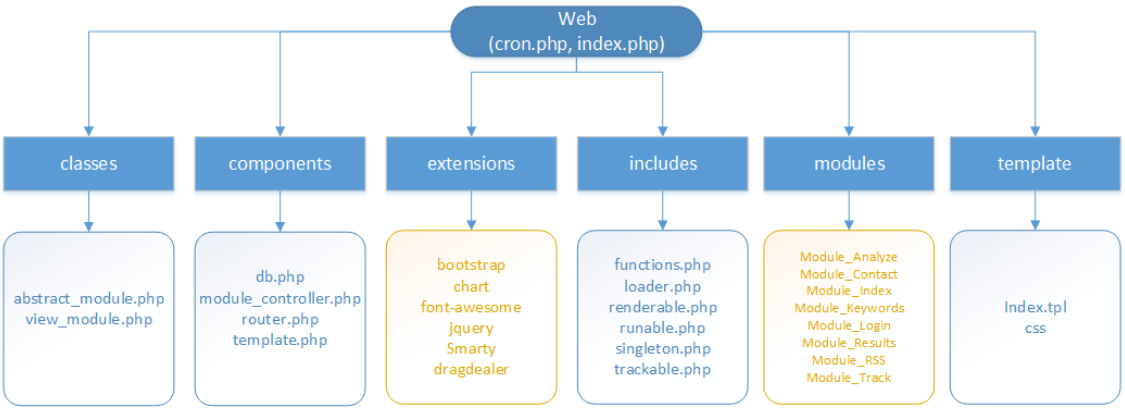
3.3 Mapa webovej stránky



Obr. 3.1: Mapa webovej stránky.

3.4 Adresárová štruktúra systému

Webová aplikácia sa skladá celkom zo šiestich knižníc (classes, components, extensions, includes, modules, template), ktoré obsahujú jednotlivé funkcie celého programu.



Obr. 3.2: Adresárová štruktúra systému.

3.5 Štruktúra programu

- **classes** – Táto knižnica obsahuje dve základné triedy, teda *Abstract_Module*, z ktorej sa dedia všetky moduly a nesie spoločné prvky všetkých modulov. Ďalšia trieda, ktorá sa tu ešte nachádza je *View_Module*, do ktorej sa ukladajú dáta pre vykreslovanie v HTML súboroch.
- **components** - Tu nájdeme 4 dôležité komponenty, ktoré zabezpečujú kontrolu nad určitými funkciami.
  Trieda *Db* slúži nato, aby sme sa pripojili k databáze s nastavenými prihlasovacími údajmi. *Module_controller* je trieda, ktorá určuje, ktorý modul bude spustený. Trieda *Router* určí obsah modulu. Trieda *template* slúži k vykresleniu stránky.
- **extensions** - Triedy, ktoré sem patria sú využívané z internetu, sú to väčšinou frameworkové knižnice. *Bootstrap* je frameworková knižnica technológie CSS. Z *Chart.js* knižnice sú využité grafy [8]. Knižnica *Font-Awesome* slúži na použité písmo [12]. *jQuery* je JavaScriptový framework, ďalej tu sa nachádza framework *Smarty*, ktorý má svoj značkovací jazyk čo prevedie TPL súbory do PHP. V poslednej rade je použitá knižnica *Dragdealer*, ktorá sa používa na posuvníky.
- **includes** – V tejto knižnici sa nachádzaju PHP súbory, ktoré je nutné, aby systém obsahoval ako prvé.
  Functions.php obsahuje funkciu *str_reverse()*, ktorá otočí text. *Loader.php* slúži na načítanie komponentov a súborov. Trieda *renderable* je abstraktná trieda, ktorú používajú moduly a šablóna, nesie sebou funkcie, ktoré sú nutné pre vykreslenie obsahu. Trieda *Runable* uchováva informácie o tom, či komponent beží alebo nie, ďalej obsahuje funkciu *run()*, ktorá slúži pre spustenie komponentov.
  Abstraktná trieda *Singleton* je tu z toho dôvodu, aby v celom systéme bežala vždy iba maximálne jedna inštancia triedy, ktorá zdedí *Singleton*. *Trackable* je rozhranie, ktoré ostatným triedám určí, že môžu použiť metódu *trackAll()* a *trackSingle()*, ktoré riešia stiahnutie dát zo vstupného zdroja. V podstate, keby bol systém napojený na viac zdrojov obsahu, napríklad Twitter, tak by aj tá trieda impelemtovala „*Trackable*“, aby ju *track* modul mohol volať.
- **modules** – V tejto knižnici sa nachádzajú jednotlivé sekcie webovej stránky, ako *Sledované stránky*, *Kľúčové slová*, *Štatistiky* a *Kontakt*.
  Modul *Module_Index* je vytvorený ako uvodná stránka webu. Modul *Module_RSS* je vytvorený pre sekciu *Sledované stránky*, ktorý načíta články pomocou RSS feedov, teda otvorí odkaz z RSS feedu. Ďalší modul *Module_Keywords* je vytvorený pre sekciu *Kľúčové slová*, ktorý slúži pre pridávanie a editácie kľučových slov, ktoré sa používajú pre vyhľadávanie článkov. Pre sekciu *Štatistiky* bol vytvorený modul *Module_Results*, ktorý slúži pre zobrazenie výsledkov analýzy z databázy.Pre samotnú analýzu bol vytvorený *Module_Analyze*, ktorý najrpv vyhľadá články podľa zadaných kľučových slov a potom tieto články prejdú cez dodaný algoritmus hodnotenia a nakoniec ich uloží do databázy a vypíše výsledky analýzy. Modul *Module_Track* slúži ako rozhranie pre spustenie sledovania článkov. Pre prihlásenie bol vytvorený *Module_Login*, kde je nastavané prihlasovacie meno a heslo.Ako posledný modul bol vytvorený pre sekciu Kontakt s názvom *Module_Contact*, kde užívateľ môže poslať správu pre správcu systému, kde potrebuje zadať iba jeho meno a jeho emailovú adresu.

- **template** – Táto knižnica obsahuje súbory so šablónou. Tu nájdeme nastavenú CSS knižnicu a *index.tpl* súbor, ktorý je základnou šablónou pre celý vzhľad stránky.

3.6 Práca so zdrojmi

Program ponúka možnosť skenovania článkov z RRS feedu [11]. Články na vybraných stránkach sa následne v nastavených intervaloch sťahujú a ukladajú do databázy, odkiaľ ich ďalej môžeme analyzovať na základe kľúčových slov.

V budúcnosti by som chcel pridať ďalšie možnosti získavania zdrojov, ale tomu sa budem venovať v nasledujúcej kapitole.

3.7 Kľúčové slová

Práve kľúčové slová tvoria hlavnú podstatu celého princípu web crawleru. Na základe užívateľom zadaných kľúčových slov algoritmus prehľadáva články, ktoré sú momentálne stiahnuté z RSS zdrojov a uložené v databáze, a týmto článkom následne pridáva hodnotenie.

Užívateľ kľúčovým slovám pri ich zadávaní tiež vyberá určité vlastnosti, ako je frekvencia sledovania a citlivosť.

3.7.1 Hľadanie kľúčových slov

Na spôsob hľadania kľúčových slov bolo zvolené SQL dotazovanie, kde sa používa príkaz *MATCH() a AGAINST()*. Napríklad, keď hľadáme presný výraz „auto“, tak algoritmus hľadá presne to slovo, pokiaľ podobné tak „aut*“ a „*uto“ s tým, že na miesto hviezdičiek môže byť čokoľvek. Aby sme vedeli vyhľadávať podobné slová, tak je potrebné do databázy uložiť celé články aj v obrátenej forme.

Je to z toho dôvodu, lebo SQL neovláda vyhľadávanie do oboch strán pomocou zástupného znaku (hviezdičky), teda nevie nájsť výraz „*uto“, dá sa nájsť „auto“ alebo „aut*“. Preto je článok otočený a hľadá sa reverzne.

3.8 CRON úlohy

CRON umožňuje periodické opakovanie procesov vo vopred stanových časových intervaloch.

V mojom programe CRON zaisťuje to, že sú každú hodinu na serveri spustené úlohy pre sledovanie článkov a hodnotenie kľúčových slov.

Algoritmus, ktorý je pomocou úlohy typu CRON spúšťaný najskôr zistí, ktoré články a ktoré kľúčové slová má analyzovať. Tento proces prebieha tak, že sa na začiatku pozrie, čo analyzoval naposledy a zistí, ktoré články ku ktorým kľúčovým slovám ešte nevyhodnotil. Potom sa pozrie na nastavenú frekvenciu sledovania pri kľúčových slovách a pri RSS zdrojoch a z nevyhodnotených článkov vyberie len tie, ktorých čas sledovania spadá do času, v ktorom prebieha tento algoritmus. Tieto články následne z RSS feedu stiahne a uloží do databázy. Nakoniec v nich prehľadá kľúčové slová a priradí im hodnotenie.

3.9 Ukladané dáta

Ako už v tejto kapitole bolo spomenuté, do databázy sú ukladané užívateľom zvolené kľúčové slová s vlastnosťami, ktoré im užívateľ nastaví (frekvencia sledovania a citlivosť, týmto vlastnostiam sa budem podrobnejšie venovať v ďalšej kapitole).

Ďalej sa ukladajú sledované stránky, ktoré si užívateľ zvolí a k nim rovnako ako pri kľúčových slovách vlastnosti, tu ich predstavuje názov stránky, frekvencia sledovania a trieda obsahu (nie je povinná), ktoré budú detailnejšie rozobraté v nasledujúcej kapitole.

Hodnotenie stiahnutých článkov na základe vybraných kľúčových slov sa získava pomocou analýzy. Toto hodnotenie je rovnako zaznamenávané do databázy, pričom je vždy priradené trojici článok, kľúčové slovo a čas.

Na základe získaného hodnotenia užívateľ môže získať štatistiky, ktoré sú automaticky vytvárané a reprezentované tabuľkou a grafmi.

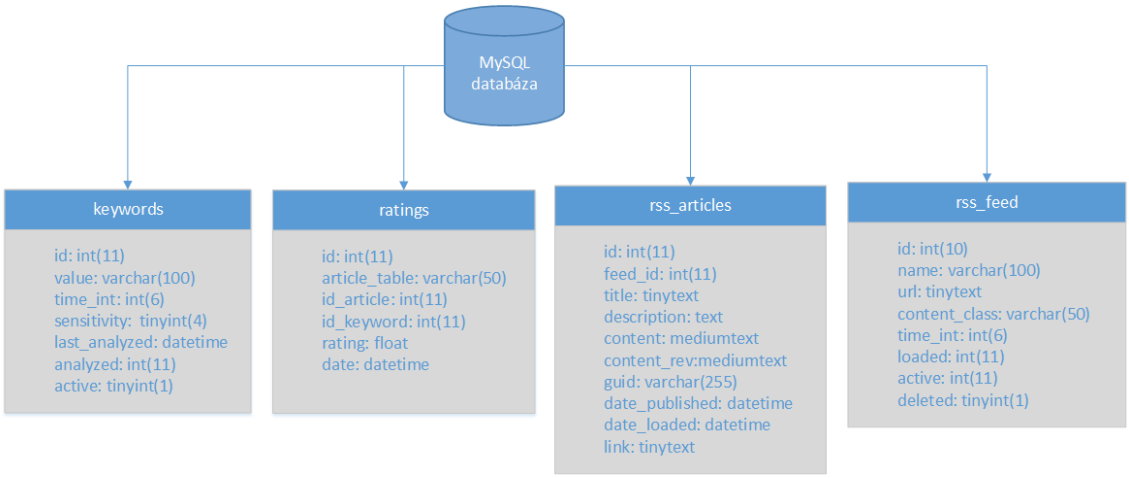
V nasledujúcich podkapitolách sa budem venovať po technickej stránke spôsobu ukladaniu a zobrazovaniu dát.

3.10 Práca s databázou

Po kľúčových slovách je ďalšou podstatou celého systému databáza, pretože spomínané časti (kľúčové slová, hodnotenie, RSS feedy, stiahnuté články z RSS feedov) systému z predošlých kapitol je nutné ukladať a načítať a práve to je podstatná funkčnosť celej databázy. V mojom prípade je systém spojený s MySQL databázou v komponente *db.php*, kde sa nachádzajú nastavené prihlasovacie údaje a funkcia *_connectToDb()*, s ktorou sa systém pripojí k databáze. Vytvorenie databázy je v samotnom súbore *peterledniczky.sql*, ktorý sa nachádza na webovom serveri.

3.10.1 Štruktúra databázy

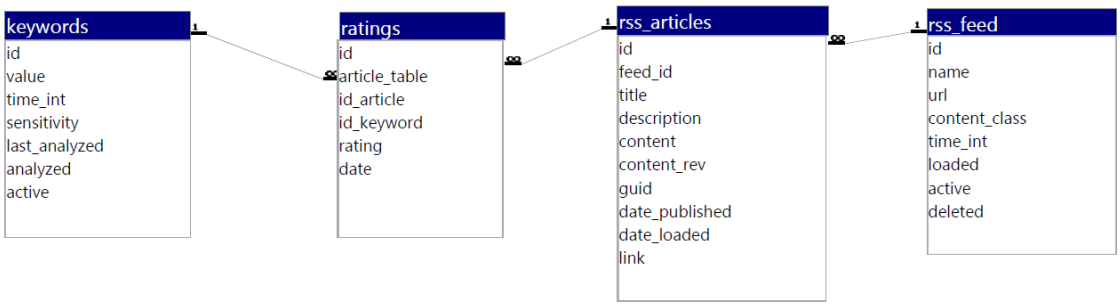
V tejto časti je ukázané ako je vytvorená databázová štruktúra systému. Na obrázku Obr. 3.3 je možné vidieť, ktoré tabuľky obsahujú jednotlivé stĺpce. Názvy jednotlivých stĺpcov sú nazvané tak, aby bolo jasné, že ktorý stĺpec aké dáta bude obsahovať a tieto dáta následne sú použité v PHP kóde, ďalej sú uvedené ešte aj dátové typy jednotlivých stĺpcov.



Obr. 3.3: Štruktúra databázy.

3.10.2 E-R diagram databázy

Na základe E-R diagramu je navrhnutá databáza systému, ktorý znázorňuje vzťah medzi tabuľkami (viz Obr. 3.4) .



Obr. 3.4: E-R diagram databázy.

3.10.3 Dátové typy databázy

Pre text je využitý dátový typ *varchar()*, *tinytext*, *text*, *mediumtext*. Pre čísla sú využité *int()*, *tinyint()*, *float* a je využitý dátový typ *datetime*.

V tabuľkách tie stĺpce, ktoré obsahujú skratku id je použitý dátový typ *int(11)* a *int(10)* , kde int značí, že v bunke daného stĺpca môže byť maximálna hodnota 2147483647, keď je signed (so znamienkom) a v prípade unsigned (bez znamienka) je to hodnota 4294967295 rozdiel medzi nimi, je v tom, že signed má rozsah od -2147483648 do 2147483647 a unsigned od 0 do 4294967295. Parameter 11 značí, že 11 číslic bude zobrazené.

Dátový typ *datetime* sa vyskytuje v stĺpcoch, kde sa nachádza výraz date, a reprezentuje aby sme mali zaznamenané, že kedy prebehla posledná analýza daného kľúčového slova, kedy sa načítali články a kedy boli publikované články.

3.10.4 Tabuľka keywords

V tejto tabuľke sú uložené kľúčové slová, ktoré chceme nájsť v článkoch.

- V stĺpci *id* sú uložené čísla, ktoré sú unikátne identifikátory elementov tabuľky, v tom prípade kľúčových slov. Ďalej sa tu nachádza stĺpec *value*, kde sú uložené názvy hľadaných kľúčových slov, dátový typ som nastavil na *varchar(100)*, kde bol nastavený parameter, že do jednej bunky môže byť vložených maximálne 100 charakterov, to znamená že maximálna dĺžka kľúčového slova môže mať 100 charakterov s medzerou vrátane.
- V stĺpci *time_int* sa nachádza nastavenie frekvencia sledovania, kde je použitý dátový typ *int(6)*, s tým že je unsigned, z toho dôvodu, lebo frekvenciu sa dá nastaviť v rozsahu od 1h do 24h, čiže nemôžu byť záporné čísla.
- Stĺpec *sensitivity* slúži k tomu aby bolo zaznamenané akou citlivosťou chceme vyhľadať kľúčové slová.
  Pokiaľ nastavíme citlivosť na *Podobný výraz* tak dostane hodnotu 0, v prípade *Presný výraz* dostane hodnotu 1 a v prípade *Presné slovo* dostane hodnotu 2.
- Do stĺpca *last_analyzed* je zaznamenaný dátum poslednej analýzy, kde je používaný dátový typ *datetime*.
- V stĺpci *analyzed* sa nachádza koľkokrát bolo dané kľúčové slovo analyzované, používa sa tu tiež dátový typ *int(11)*.
- Posledný stĺpec je *active*, kde je zaznamenané to, či dané kľúčové slovo má hľadať systém alebo nie. Sú tu uložené iba dve hodnoty. Hodnota 1 je v tom prípade, keď je aktívne a 0 keď nie. Je tu využitý dátový typ *tinyint(1)*, pretože rozsah je od 0 do 1, a do tohto typu vieme uložiť 1 byte.

3.10.5 Tabuľka ratings

Táto tabuľka slúži nato, aby sme pridali hodnotenie k jednotlivým článkom.

- V stĺpci *article_table* je určené, že k čomu patrí hodnotenie, čiže obsahuje názov tabuľky, s ktorou má byť spojený.
- Ďalší stĺpec *id_article* odkazuje na tabuľku *rss_articles* a obsahuje pole, v ktorom je identifikátor nájdeného článku.
- *Id_keyword* je načítané z tabuľky *keywords*, čo značí identifikátor kľúčového slova.
- Stĺpec *rating* obsahuje hodnotenie kľúčového slova v článkoch, sú to náhodne generované čísla a používam tu dátový typ *float*, z toho dôvodu lebo v štatistikách sú zobrazené priemerné hodnotenie, kde sú desatinné čísla.

3.10.6 Tabuľka rss_articles

V tejto tabuľke sa nachádzajú stiahnuté články, ktoré našla aplikácia na pridaných RSS feedoch.

- Stĺpec *feed_id* odkazuje na tabuľku *rss_feed* a obsahuje číslo, ktoré značí identifikátor RSS feedu.
- Stĺpce *title*, *description*, *guid* a *link* obsahujú obsahy, ktoré sú stiahnuté priamo z RSS feedov.
- V stĺpci *conent* a *content_rev* sú uložené celé stiahnuté články. V prípade *content_rev* je celý článok otočený, z toho dôvodu, aby systém dokázal vyhľadať aj podobné alebo presné výrazy.

3.10.7 Tabuľka rss_feed

Tu sú uložené RSS feedy webových stránok, odkiaľ sú stiahnuté články.

- V stĺpci *name* sú uložené zvolené názvy RSS feedov.
- V stĺpci *url* sa nachádzajú RSS adresy, na ktorých prebieha analýza.
- Do stĺpca *content_class* sa ukladá trieda obsahu, kde máme možnosť vyhľadať v presných tagoch obsah.
- V stĺpci *loaded* je uložené, koľkokrát bol feed načítaný.

3.11 Štatistiky kľúčových slov

V systéme nájdeme štatistiky kľúčových slov, ktoré sú zobrazené na obrázku Obr. 3.5. Kde počet článkov značí, že koľkokrát sa našlo dané kľúčové slovo v článkoch. Na spočítanie článkov bola využitá funkcia *COUNT()* . Každý článok je hodnotený iba jeden krát, a počet hodnotení je vlastne počet článkov pre dané kľúčové slovo, čiže je spočítané počet hodnotení.

Priemerné hodnotenie, ktoré je možné vidieť na obrázku Obr. 3.5 pre dané kľúčové slovo funguje tak, že systém vyhodnotí tie články, ktoré našiel ten daný deň potom každé kľúčové slovo, ktoré sa vyskytovalo v článkoch uloží do databázy a dostane hodnotenie pomocou generovanie náhodných čísel v rozsahu od 1 do 10 a je zobrazené priemerné hodnotenie pomocou funkcie *AVG()* .

Štatistiky kľúčových slov

Slovo	Interval sledovania	Citlivosť	Článkov	Priemerné hodnotenie	Dátum poslednej analýzy
česko	12 h	Podobný výraz	538	5.69	31.05.2016 12:04:06
slovensko	12 h	Presný výraz	398	5.45	31.05.2016 12:04:06
Most	12 h	Presné slovo	1064	5.5	31.05.2016 12:04:07
práce	12 h	Podobný výraz	1364	5.44	31.05.2016 12:04:07

Obr. 3.5: Štatistiky kľúčových slov.

3.12 Práca so Smarty

Šablónovací systém Smarty je využitý z toho dôvodu, aby bola rozdelená prezentačná vrstva od logickej vrstvy. Aby sme vedeli oddeliť tieto dve vrstvy od seba, je potrebné vytvoriť minimálne dva súbory jeden pre šablónu a jeden pre programátorské účely.

Pre vytváranie šablóny pre šablónovací jazyk Smarty sa používa súbor *.tpl*, v ktorom sa nachádzajú všetky kódy, ktoré sa týkajú prezentačnej vrstvy, teda ako napríklad v mojom prípade html kód, css a javascript.

Pre programátorské účely je nutné vytvoriť súbor PHP, kde pre prechod medzi súbormi PHP a šablónovacími súbormi sú vytvorené špeciálne Smarty premenné. S príkazom *assign* vytvoríme premenné pre šablónovací jazyk Smarty, kde určíme, ktorý PHP premenné bude Smarty premenné a ďalej v šablóne iba odkazujeme na vytvorenú premennú pomocou symbolu *$* a názvom, ktorý bol vytvorený pre Smarty.

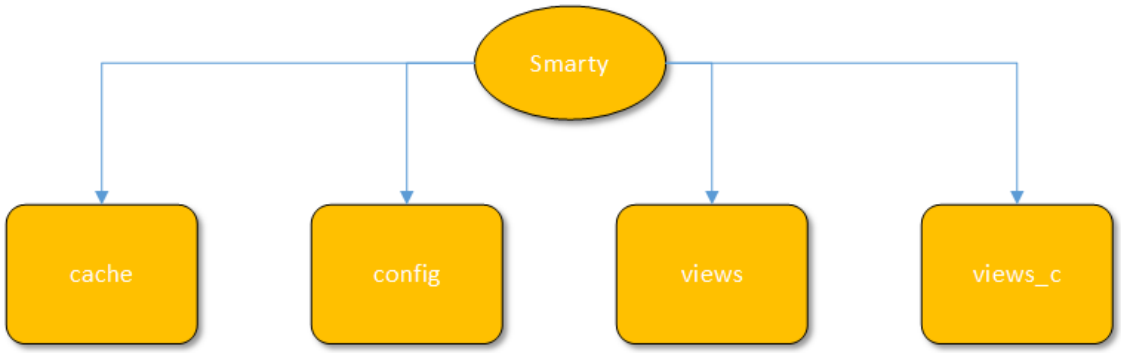
Príkaz, ktorý je ešte nutné aby PHP kód obsahoval je *display*, kde určíme súbor, v ktorom chceme zobraziť obsah vytvoreného premenného.

3.12.1 Adresárová štruktúra pri využití

Pri generovaní obsahu šablónovací jazyk Smarty [6] používa 4 základné adresáre (viz Obr. 3.6). Tieto adresáre sú zvyčajne koreňom webovej stránky.

Adresár *cache* a *views_c* sú určené pre kompilované *.tpl* súbory, čiže slúžia ako medzipamäť. Táto možnosť je dobrá k tomu, aby server nebol zaťažený, pretože v medzipamäti sú len HTML kódy.

Adresár *views* a *config* sú určené pre šablóny, ktoré vytvárame my. Tu sa nachádzajú *.tpl* súbory pre vytváranie šablón a konfiguračných súborov.

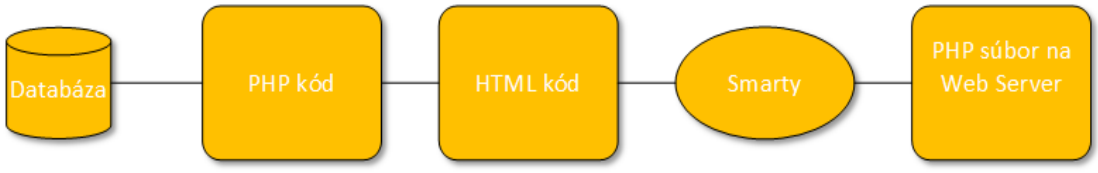


Obr. 3.6: Adresárová štruktúra jazyka Smarty.

3.12.2 Vykreslenie obsahu

Vykreslenie obsahu (viz Obr. 3.7) prebieha tak, že v PHP kóde zadáme, že ktoré dáta potrebujeme zobraziť z MySQL databázy pomocou premennou *$query* a samozrejme je potrebné ešte naprogramovať funkcie, ktoré spočítajú v mojom prípade počet nájdených článkov na dané kľúčové slovo alebo priemerné hodnotenie.

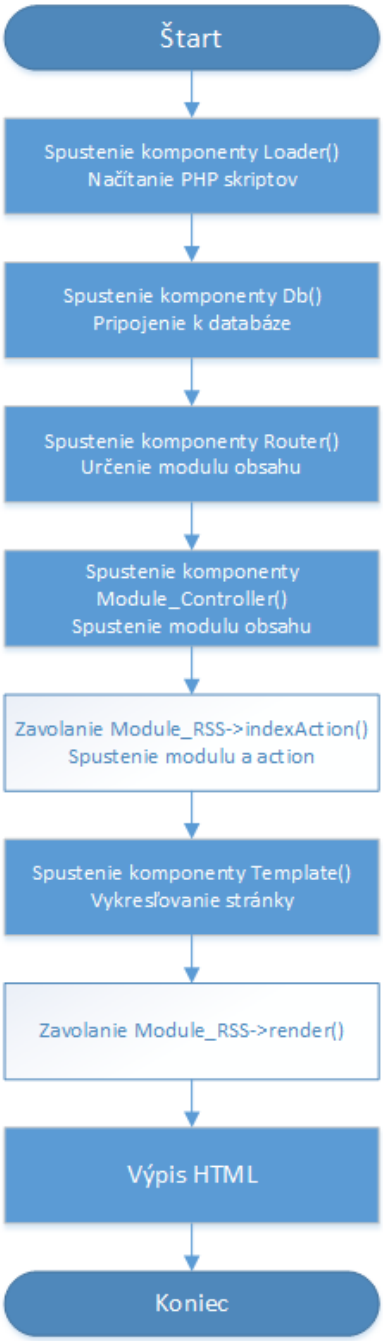
V šablónovacom súbore *.tpl* zobrazíme správnu premennú, ktorá bola vytvorená v PHP kóde pre Smarty. Potom šablónovací systém Smarty prevedie *.tpl* súbory na PHP súbor, ktorý sa dostane na Webový server čo spustí kód a následne sa obsah vykresľuje.



Obr. 3.7: Vykreslenie obsahu pomocou Smarty.

3.13 Schéma funkčnosti systému

Z programátorského hľadiska na schéme Obr. 3.8 je znázornená funkčnosť systému v sekcii RSS pre pochopenie celého systému. Spustia sa komponenty, ktoré obsahujú jednotlivé činnosti ako načítanie PHP skriptov, pripojenie k databáze, určenie a spustenie modulu obsahu, vykresľovanie stránky a ako posledný krok je výpis HTML stránky.



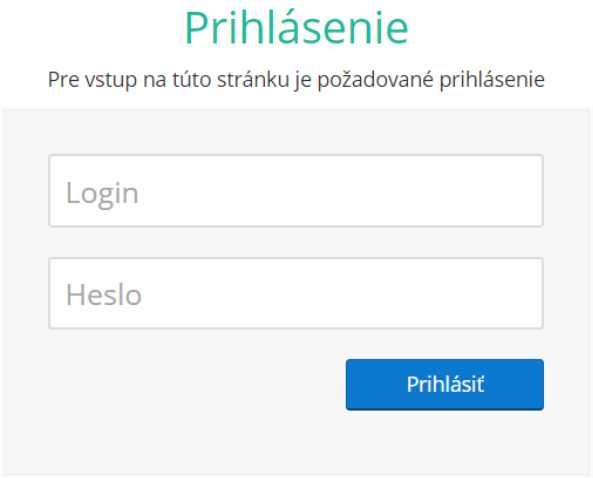
Obr. 3.8: E-R diagram funkčnosti systému.

4 PRÁCA S PROGRAMOM

V tejto kapitole popisujem spôsob práce s programom z užívateľského pohľadu čo by mohlo slúžiť ako manuál k celému systému. Cieľom je priblížiť potenciálnym užívateľom, ako program plnohodnotne používať a získavať z neho informácie, o ktoré majú záujem.

4.1 Prihlásenie do užívateľského rozhrania

Keď užívateľ otvorí webovú stránku tak vyskočí okno (viz Obr. 4.1), kde od neho systém vyžaduje prihlasovacie údaje, ktoré mu budú pridelené pred použitím. Z bezpečnostných dôvodov je nutné toto riešenie, aby boli hackerské útoky minimalizované.



Obr. 4.1: Prihlasovacie okno do systému.

4.2 Pridanie stránky pre sledovanie

Po úspešnom prihlásení má užívateľ na domácej stránke možnosť pridať stránku pre sledovanie. Po kliknutí na túto možnosť sa mu otvorí záložka *Sledované stránky*, kde má možnosť pridať novú stránku alebo pozmeniť stránky už sledované (viz Obr. 4.2).

Celý proces sledovania je založený na RSS feedoch. RSS feed umožňuje odber článkov z vybraných serverov [11]. Užívateľ si teda môže zvoliť, ktoré stránky ho zaujímajú a aké informácie chce získavať a ďalej analyzovať.

Obr. 4.2: Pridanie stránky pre sledovanie.

4.2.1 Názov stránky

Do riadku Názov stránky užívateľ zadá názov, pod ktorým bude daná stránka uložená v databáze a následne zobrazovaná v štatistikách (viz Obr. 4.2). Hlavnou úlohou je prehľadnosť pre užívateľa programu. Ako príklad uvediem názov *Zprávy iDNES.cz*.

4.2.2 RSS URL

Riadok RSS URL slúži pre zadanie RSS linku stránky, ktorú chceme sledovať (viz Obr. 4.2). Túto URL užívateľ získa zo stránky, ktorú chce sledovať. Väčšinou býva uvedená priamo na hlavnej stránke a označená príslušným symbolom. Ako príklad uvediem nasledujúcu RSS URL *http://servis.idnes.cz/rss.aspx?c=zpravodaj* .

4.2.3 Trieda obsahu

Trieda obsahu je voliteľná položka, ktorá dáva užívateľovi možnosť napísať presný tag, teda presnú HTML značku (tag), v ktorom sa nachádza obsah článku v danom RSS feede (viz Obr. 4.2) .

Túto vlastnosť som zaradil vzhľadom k tomu, že webové stránky nemajú pevnú štruktúru, preto nie je jednoduché zostaviť univerzálny algoritmus, ktorý nájde článok stopercentne na každej webovej stránke. Najčastejšie sa článok vyskytuje medzi tagmi <ARTICLE>…</ARTICLE> , ale nie je to pravidlo.

Aktuálne mám vytvorený jednoduchý algoritmus, ktorý dokáže inteligentne vyhľadávať na webových stránkach a aj v prípade, že nenájde tag <ARTICLE>, pokúsi sa napriek tomu článok nájsť. Avšak, ako som sa už zmienil, momentálne je používaný pomerne jednoduchý algoritmus, z tohto dôvodu sa niekedy pri netypických RSS feedoch nepodarí obsah článku nájsť. To je dôvod, prečo som sa rozhodol do programu zaradiť položku Trieda obsahu, kde môže užívateľ priamo zadať tag, v ktorom má algoritmus článok hľadať.

V budúcnosti by som chcel tento problém odstrániť a algoritmus vylepšiť. Inšpiráciu by som hľadal pri čítačke článkov z prehliadača SAFARI, ktorá si dokáže poradiť s akýmkoľvek RSS feedom. Podrobnejšie sa týmto vylepšením budem zaoberať v poslednej kapitole Návrhy do budúcnosti.

4.2.4 Frekvencia sledovania

Užívateľ si môže nastaviť frekvenciu, s akou chce články sťahovať (viz Obr. 4.2). Pre program táto informácia znamená, ako často bude pomocou úloh typu CRON spúšťané sťahovanie článkov z vybraného zdroja.

Frekvencia sledovania je v tomto prípade veľmi dôležitou položkou, vzhľadom k tomu, že z RSS feedov je možné stiahnuť len posledných 10 alebo 20 článkov, teda nie je možné stiahnuť všetky články, ktoré sa nachádzajú na serveri. To znamená, že od frekvencie je závislá kompletnosť článkov zo zdroja. To napríklad znamená, že keby si užívateľ nastavil frekvenciu sledovania na 24 hodín a za jeden deň by na danom RSS feede pribudlo 50 článkov, tak do databázy sa nestiahnu všetky tieto články, ale len tie najnovšie, pretože staršie sa nimi nahradia.

Na druhej strane, ak si užívateľ nastaví frekvenciu sledovania na každú hodinu, tieto články zaťažia server, čo spôsobí dlhú odozvu programu.

Z tohto dôvodu je veľmi dôležité frekvenciu sledovania správne nastaviť, aby nedošlo k nežiadúcim účinkom.

4.2.5 Úprava stránky pre sledovanie

Ďalšou možnosťou pre užívateľa na záložke *Sledované stránky* je úprava už sledovaných stránok. Táto možnosť sa spustí po kliknutí na Názov alebo URL adresu danej stránky v zozname.

Je tu možné meniť všetky nastavenia stránky (viz Obr. 4.3), teda názov, RSS URL, triedu obsahu ako aj frekvenciu sledovania. Ďalej je tu tiež možnosť sledovanie zapnúť alebo vypnúť.

Obr. 4.3: Úprava stránky pre sledovanie.

4.3 Pridanie kľúčového slova

Po otvorení webovej stránky je na domovskej stránke možnosť pridať kľúčové slovo. Po kliknutí na túto možnosť sa otvorí záložka *Kľúčové slová*, na ktorej sa nachádza zoznam kľúčových slov, ktoré už boli užívateľom pridané.

Kľúčovými slovami užívateľ špecifikuje oblasť záujmu, teda ich zvolí podľa problematiky, ktorú chce sledovať. Toto slovo sa ďalej bude vyhľadávať v stiahnutých článkoch, ku ktorým bude toto slovo pridelené.

Obr. 4.4: Pridanie kľúčového slova.

4.3.1 Kľúčové slovo

Do riadku kľúčové slovo užívateľ zadá slovo, ktorého prítomnosť chce prehľadávať v článkoch stiahnutých v databáze, môže si tiež vybrať, v ktorých článkoch bude toto slovo prehľadávané (viz Obr. 4.4).

4.3.2 Frekvencia sledovania

Pri kľúčových slovách funguje frekvencia sledovania rovnako ako pri RSS feede, ale je to len dočasné riešenie, aby som mohol sledovať, nakoľko táto činnosť zaťažuje server (viz Obr. 4.4). Teoreticky by sa dalo nastaviť, aby server analyzoval kľúčové slová v článkoch napríklad každý deň okolo tretej hodiny ráno, keď nie je vyťaženie servera.

Na druhej strane, ak bude mať užívateľ nastavených okolo 100 kľúčových slov a za deň bude na server stiahnutých okolo 1000 článkov, je potrebné, aby bol analyzovaný každý článok ku každému kľúčovému slovu. To by znamenalo 100 000 operácií a je otázne, či by takýto výkon server zvládol a nedošlo by k problémom s jeho preťažením.

Z tohto dôvodu by som na začiatku navrhoval sledovať výkonnosť servera pri vykonávaní takýchto operácií. Následne na základe tejto analýzy stanoviť jeden interval analýzy kľúčových slov tak, aby bol dopad na výkonnosť a funkčnosť servera čo najmenší. Teda by sa táto možnosť nastavenia frekvencie sledovania pri kľúčových slovách užívateľovi znemožnila a server by túto analýzu vykonával v stanovenom čase, v ktorom by bol server optimálne zaťažený.

4.3.3 Citlivosť

Ďalej môže užívateľ nastaviť kľúčovému slovu citlivosť (viz Obr. 4.5), to znamená presnosť, s akou sa bude daný výraz v článkoch vyhľadávať. Užívateľ má 3 možnosti nastavenia úrovne citlivosti vyhľadávania. Môže vybrať podobný výraz, presný výraz alebo presné slovo. Rozdiel medzi jednotlivým úrovňami citlivosti vysvetlím na nasledujúcom príklade.

Ak si zvolím ako moje kľúčové slovo „slniečko“ a zvolím citlivosť na úrovni presné slovo, nájde mi v článku len presne tento tvar slova, takže slová „slniečka“, „slniečkom“, „slniečkami“ a podobne, v článku nenájde. Tieto slová algoritmus nájde v prípade, že citlivosť bude nastavená na úrovni podobný výraz, čo znamená, že bude prehľadávať rôzne koncovky daného kľúčového slova, pričom základ slova zostane nezmenený.

Obr. 4.5: Citlivosť.

Ďalej si zvolím ako kľúčové slovo „auto“, ako to bolo aj pri predchádzajúcom príklade, pri citlivosti na úrovni presné slovo bude nájdené len konkrétne slovo v tomto tvare. Avšak pri citlivosti na úrovni presný výraz bude hodnotená pozitívne prítomnosť kľúčového slova aj pri výrazoch ako „automat“, „automobil“ a tak ďalej. Čo môžem zhodnotiť tak, že citlivosť na úrovni presný výraz bude hľadať kľúčové slovo ako základ slova s rôznymi príponami a predponami.

4.4 Analýza

Analýza v tomto programe znamená hodnotenie článkov vzhľadom ku kľúčovým slovám na základe náhodne generovaných čísiel. Na vstupe má tento algoritmus konkrétny článok a výstupom je číslo, hodnota, ktorú článku priradí. Táto analýza je spúšťaná automaticky prostredníctvom CRON úloh v nastavených časových intervaloch.

Okrem prednastavených CRON úloh môže užívateľ hodnotenie jednotlivých kľúčových slov spustiť aj ručne. Táto možnosť je momentálne stále vo vývoji, vzhľadom k jej výkonovej náročnosti. Preto je ešte potrebné tento proces testovať.

Ručné spustenie analýzy slov som do programu zaradil z toho dôvodu, že v budúcnosti, keď bude v databáze veľký počet stiahnutých článkov za obdobie napríklad dvoch a viac rokov a užívateľ sa rozhodne pridať nové kľúčové slovo a bude ho z nejakého dôvodu potrebovať okamžite analyzovať, môže túto analýzu spustiť hneď v danom momente. Avšak pri takom veľkom počte článkov by bola táto analýza veľmi výkonovo náročná, z tohto dôvodu je táto služba skôr zámerom pre vylepšovanie programu v budúcnosti.

5 VÝSTUPY A VYUŽITIE PROGRAMU

5.1 Výstupy

Výstupom programu sú štatistiky údajov, ktoré boli zhromaždené za dlhší časový horizont (niekoľko mesiacov, rokov) jedným kľúčovým slovom. Počas tohto dlhšieho časového horizontu sú v databáze zhromažďované články z rôznych stránok, v ktorých sa nachádza dané kľúčové slovo. Aktuálne sú články hodnotené na základe nálady článku, čo je riešené pomocou generovanie náhodných čísiel a toto hodnotenie je rovnako zaznamenané v databáze. Po istom čase, keď bude vzorka pre užívateľa dostatočne reprezentatívna, môže toto kľúčové slovo štatisticky vyhodnotiť. Výstupom štatistického vyhodnotenia sú grafy (viz Obr. 5.1, Obr. 5.2).

5.2 Využitie

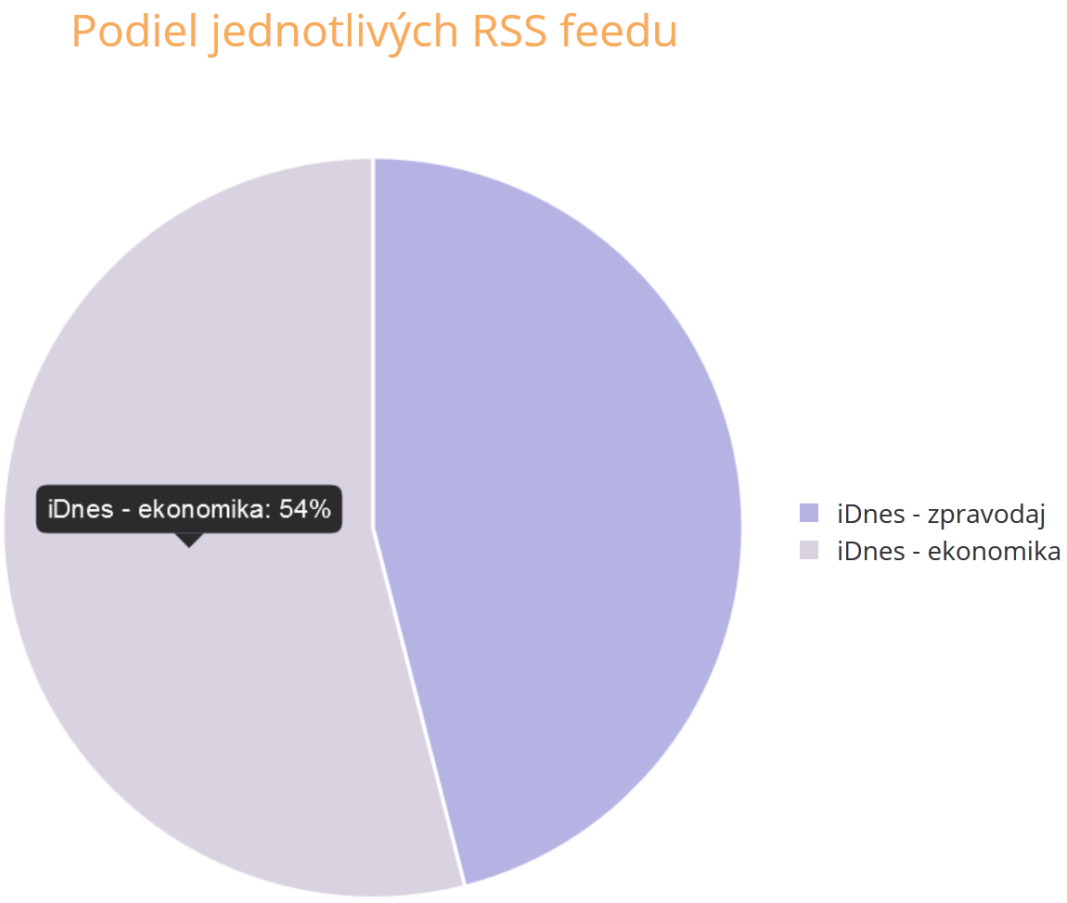
Jedno z možných využití vidím v grafickom vykreslení nálady kľúčového slova v čase (viz Obr. 5.1). Toto zobrazenie je možné využiť pri analýze politických alebo ekonomických situácií.

Obr. 5.1: Využitie.

Ďalším možným využitím by mohlo byť sledovanie činnosti konkrétnej politickej strany na serveroch zaoberajúcich sa politickými témami. Užívateľ by do RSS feedu zadal servery, ktoré sa venujú politickej situácii a aktuálnym témam z tejto oblasti a ako kľúčové slovo by zvolil názov niektorej konkrétnej politickej strany, o ktorej výsledky by mal záujem. Týmto spôsobom môže zistiť, ako sa vyvíjali názory na danú politickú stranu počas dlhšieho časového obdobia a ako sa strane darilo. Prípadne by mohol týmto spôsobom porovnať činnosť viacerých politických strán v jednom období, a tým získať prehľad o politickej scéne v danom čase.

Obmenou predchádzajúceho príkladu by mohlo byť širšie sledovanie politickej situácie. Užívateľ by do sledovania zadal viac serverov, ktoré sa venujú politike a ako kľúčové slová by zvolil všetky hlavné politické strany. Následne môže analyzovať, ako jednotlivé servery hodnotia politické strany. Výsledkom by mohlo byť sledovanie výkyvov od priemeru, ktoré by ukazovali, aký postoj majú servery k jednotlivým stranám, či sa v článkoch vyskytujú náznaky nadržiavania konkrétnej strane, prípadne zhadzovanie inej politickej strany.

Ďalším typom analýzy by mohol byť počet článkov s určitým kľúčovým slovom na rôznych RSS feedoch. Výstupom tejto analýzy by bol nasledujúci graf (viz Obr. 5.2). Interpretáciu by som videl v záujme daného média o zvolenú tému.



Obr. 5.2: Podiel jednotlivých RSS feedov.

Podobne ako na politickú situáciu, by bolo možné aplikovať analýzy napríklad na nadnárodné firmy, kde by bolo rovnako možné sledovať názory, ktoré sa vyskytujú v článkoch, teda aký názor má verejnosť na danú spoločnosť. Prípadne by bolo možné firmu sledovať v dlhšom časovom období a získať graf jej vývoja, ako sa firme darilo, aké mala postavenie na trhu. Ďalej by sa dalo sledovať situáciu v školstve, vývoj a počet reforiem vzdelávacieho systému, prípadne názor verejnosti na tieto reformy.

Podobnú analýzu by bolo možné aplikovať na akýkoľvek sektor, napríklad zdravotníctvo, hospodárstvo, kultúru, a tak ďalej.

5.2.1 Využitie z ekonomického hľadiska

Aktuálna verzia systému ponúka možnosť prihlásenie iba pre administrátora, z toho dôvodu aby systém bol uzamknutý. Pretože inak by mohol pridávať a odoberať kľúčové slová, ako aj sledované stránky ktokoľvek, teda mohol by nastať problém pri tom, keby jeden užívateľ zmazal slová pridané iným užívateľom.

Z tohto dôvodu, by som v budúcnosti navrhoval umožniť tieto funkcie iba pre registrovaných užívateľov, ktorí by si po prihlásení na vlastné konto mohli každý zvoliť vlastné stránky, z ktorých by chceli čerpať informácie, ako aj kľúčové slová, ktoré ich zaujímajú. Zobrazenie výsledkov štatistík užívateľom, to znamená grafov a tabuliek, by som navrhoval spoplatniť nejakou menšou finančnou čiastkou tak, aby bol server sebestačný. To znamená, že by si touto službou vedel zarobiť finančné prostriedky na vlastnú prevádzku, a rovnako aj na prípadné inovácie.

6 NÁVRHY DO BUDÚCNOSTI

Ako prvý návrh do budúcnosti by som chcel pridať externý algoritmus, ktorý by bol schopný určiť hodnotenie článkov vzhľadom ku kľúčovým slovám.

Ako som už spomínal v predchádzajúcej kapitole, ďalším z návrhov do budúcnosti by som chcel spoplatniť zobrazenie výsledkov analýz. K tomuto kroku by som však pristúpil až neskôr, po dvoch až troch rokoch prevádzky, keď bude na serveri väčšia databáza článkov, kľúčových slov a histórie ich hodnotenia. Istý čas by som počkal z toho dôvodu, aby som mal zákazníkom čo ponúknuť, a aby sa im oplatilo za tieto služby zaplatiť.

Ďalším možným návrhom by bolo rozšírenie spôsobu získavania článkov. Momentálne sú články získavané len prostredníctvom RSS. Na začiatok by som pridal skenovanie twitterových príspevkov známych osobností pôsobiacich napríklad v politickej alebo ekonomickej oblasti.

Potom by som navrhoval ďalšie možnosti rozšírenia, ako je naprogramovanie Web Bota, ktorý bude sám neustále skenovať webové stránky a hľadať na nich požadované kľúčové slová. Touto možnosťou by sa užívateľovi uľahčila práca s programom, zadával by len kľúčové slová, o ktoré má záujem, odpadla by mu starosť so zadávaním RSS URL zdrojov, ktoré chce skenovať. Zároveň by táto možnosť rozšírila databázu skenovaných stránok, teda aj zaznamenaných článkov. Program by tak mal omnoho širšie využitie a spektrum užívateľov a najmä analýzy by boli oveľa presnejšie, vzhľadom k väčšej vzorke článkov.

Nevýhodou tohto riešenia je vysoké využitie serveru pre Web Bota. Bolo by potrebné zakúpiť alebo prenajať si skutočne výkonný server. Momentálne, na začiatku tohto projektu, by bol problém nájsť na toto opatrenie finančné prostriedky. Riešením by mohli byť poplatky za zobrazovanie výsledkov štatistík, ktoré som už spomínal, alebo by bolo vhodné nájsť investora, ktorý by projekt finančne podporil.

Tiež by som chcel v budúcnosti vyriešiť problém s čítaním RSS feedov a hľadaním článkov vo feede, ktorý má atypickú štruktúru. Ako som už spomínal, pri riešení tohto problému by som sa rád inšpiroval čítačkou článkov z prehliadača Safari, ktorá si dokáže poradiť s akýmkoľvek RSS feedom. Princíp tejto čítačky je založený na predpoklade, že dôležitý obsah je umiestnený v tagoch <ARTICLE>, <DIV> alebo <SPAN>, nezáleží na tom, v ktorom z nich, kým to nie je v tagu <P>. Ďalej prehliadač meria dĺžku obsahu, ten musí byť dostatočne dlhý, každý odstavec by mal mať aspoň 100 znakov.

Ďalej by som chcel vylepšiť ručné spúšťanie analýzy kľúčových slov, ktoré je aktuálne vo vývoji a testuje sa jeho výkonová náročnosť. Zatiaľ je využívaná analýza len na základe prednastavených CRON úloh, ktoré sa spúšťajú v určitých časových intervaloch. Ako som už spomínal v predchádzajúcich kapitolách, funkcia ručného spúšťania analýzy by mohla mať pre užívateľov veľký význam hlavne pri pridaní nového kľúčového slova a potrebe jeho okamžitého zanalyzovania vo všetkých článkoch v databáze. Problém s výkonovou náročnosťou tejto úlohy na server by som musel riešiť hlavne po pár rokoch, keď by bola databáza už veľmi široká a počet článkov, ktoré by bolo potrebné prehľadať by bol pre server zaťažujúci. Túto funkciu by bolo takisto dobré spoplatniť kvôli zaťaženiu servera.

7 ZÁVER

Cieľom bakalárskej práce bolo zoznámiť sa s problematikou návrhu webových aplikácii, navrhnúť aplikáciu, ktorá bude vyhľadávať z iných stránok záznamy, kde sa vyskytuje zmienka o hľadanom produkte. Navrhnúť vhodné grafy a HTML stránky, ktoré budú zobrazovať trend vývoja nálady ohľadom daného produktu.

Bakalárska práca obsahuje popis funkčnosti programu z technického ako aj užívateľského hľadiska. Práca s programom je užívateľsky nenáročná, na vstupe užívateľ zadáva zdroje, z ktorých by chcel čerpať články a kľúčové slová, ktoré špecifikujú oblasť jeho záujmu. Program sa postará o zhromažďovanie článkov do MySQL databázy v pravidelných časových intervaloch a o vyhľadávanie kľúčových slov. Ďalej do databázy ukladá k článkom hodnotenie, ktoré prebieha na základe algoritmu generovanie náhodných čísiel v rozsahu od 1 do 10 a predstavuje náladu, ktorá je článku priradená.

Následne má užívateľ možnosť štatistickej analýzy nazbieraných dát, ktorých výstup predstavujú grafy. Grafy môžu znázorňovať vývoj nálady daného kľúčového slova v určitom časovom období, alebo tiež počet článkov s daným kľúčovým slovom v závislosti na podiele nálady.

Ďalej som v práci tiež uviedol možnosti implementácie programu do rôznych oblastí, ako sú ekonómia a politická situácia. V týchto odvetviach je možné časové sledovanie nálady určitého objektu alebo tiež sledovanie odchýlok od priemeru. Rovnako je možné program aplikovať na iné segmenty, využitie by bolo podobné, rozdiel by bol vo výpovednej hodnote a interpretácii analýzy.

Na záver by som rád spomenul návrhy do budúcnosti, na ktoré som sa v práci takisto zameral. Program by som chcel ďalej vyvíjať ako po technickej stránke, tak aj po stránke užívateľskej. Prvým krokom by bolo vytvorenie administratívneho rozhrania, teda umožnenie registrácie ďalších užívateľov. Ďalej by som sa venoval rozšíreniu vyhodnotení článkov, možností získavania článkov, ako aj novej možnosti spustenia analýzy článkov ručne užívateľom.

LITERATURA

[1] DUCKETT, J. *HTML & CSS: design and build websites*. Indianapolis, IN: Wiley, 2011. ISBN 1118008189.

[2] NIXON, R. *Learning PHP, MySQL, JavaScript, and CSS*. 2nd ed. Sebastopol, CA: O'Reilly, 2012. ISBN 1449319262.

[3] HTML(5) Tutorial: Examples in Every Chapter. In: *W3schools* [online]. W3.CSS, 2016 [cit. 2016-05-31]. Dostupné z: http://www.w3schools.com/html/default.asp

[4] CSS Tutorial: Examples in Each Chapter. In: *W3schools* [online]. W3.CSS., 2016 [cit. 2016-05-31]. Dostupné z: http://www.w3schools.com/css/

[5] PHP: Hypertext Preprocessor. In: *Php: PHP 5.6.22* [online]. The PHP Group, 2016 [cit. 2016-05-31]. Dostupné z: http://php.net

[6] PHP Template Engine Smarty: Smarty 3 Documentation. In: *Smarty* [online]. New Digital Group, Inc., 2016 [cit. 2016-05-31]. Dostupné z: http://www.smarty.net/documentation

[7] MySQL: MySQL Documentation. In: *MySQL* [online]. Oracle Corporation, 2016 [cit. 2016-05-31]. Dostupné z: http://dev.mysql.com/doc/

[8] Chart.js: Getting started. In: *Chartjs* [online]. 2016 [cit. 2016-05-31]. Dostupné z: http://www.chartjs.org/docs/

[9] JavaScript Tutorial: Examples in Each Chapter. In: *W3schools* [online]. W3.CSS, 2016 [cit. 2016-05-31]. Dostupné z: http://www.w3schools.com/js/

[10] Bootstrap. In: *Geteboostrap* [online]. MIT [cit. 2016-05-31]. Dostupné z: http://getbootstrap.com/

[11] XML RSS: RSS Document Example. In: *W3schools* [online]. W3.CSS, 2016 [cit. 2016-05-31]. Dostupné z: http://www.w3schools.com/xml/xml_rss.asp

[12] Font Awesome: Font Awesome CDN. In: *Fontawesome* [online]. CC BY 3.0 [cit. 2016-05-31]. Dostupné z: http://fontawesome.io/get-started/

[13] jQuery Tutorial: "Try it Yourself" Examples in Each Chapter. In: *W3schools* [online]. W3.CSS, 2016 [cit. 2016-05-31]. Dostupné z: http://www.w3schools.com/jquery/

ZOZNAM SYMBOLOV, VELIČÍN A SKRATIEK

HTML	HyperText Markup Language
CSS	Cascading Style Sheets
PHP	Hypertext Preprocessor
SQL	Structured Query Language
CRON	Command Run On
RSS	Rich Site Summary
tpl	Template
E-R	Entity–Relationship

ZOZNAM PRÍLOH

A INŠTRUKCIE K INŠTALÁCIE

1. Obsah priečinku web, ktoré nájdete na priloženom CD skopírujte na webový server, tu sú súbory celého systému.
2. Vytvorte prihlasovacie meno a heslo k databáze, ktoré nastavíte v súbore *web\components\db.php*
3. Z priloženého CD skopírujte súbor *sql\peterledniczky.sql* do databázy, ktorá je na vašom webovom serveri. S týmto súborom vytvoríte a nastavíte databázu.
4. Prihlasovacie meno a heslo je treba nastaviť v súbore *web\modules\login\login.php*.

B POŽADOVANÁ KONFIGURÁCIA

1. Web server Apache verzia 2.0 a vyššia
2. PHP 5.6
3. MySQL 5.0

C SÚBORY NA CD

1. **Bakalárska práca** – Tento priečinok obsahuje hlavný dokument bakalárskej práce vo formáte PDF.
2. **web –** Tento priečinok obsahuje všetky zdrojové súbory programu
3. **sql** – Tento priečinok obsahuje súbor *peterledniczky.sql*, ktorý je určený pre vytvorenie databázy.